Faculdade de Ciência e Tecnologia
Engenharia de Elétrica
Disciplina: Eletrônica Digital
Professor: Vitor Leão Filardi



Apostila de Eletrônica Digital

Sumário

Referências Bibliográficas

IDOETA, I. V.; CAPUANO, F. G. *Elementos de Eletrônica Digital*. [S.l.]: Editora Érica, 1984.

IDOETA, I. V.; CAPUANO, F. G. *Sistemas Digitais-Princípios e Aplicacoes*. [S.l.]: Editora Érica, 1984.

Capítulo 1

Primeira Unidade

1.1 Sistema de Numeração

0 1 2 3 4 5 6 7 8 9 → Decimal

2003 → 2000 + 000 + 00 + 3
$2 * 10^3 + 0 * 10^2 + 0 * 10^1 + 3 * 10^0$
abc= $a * 10^2 + b * 10^1 + c * 10^0$

1.1.1 Polinômio Geral

$(n)_b = n_i * b^i + n_{i-1} * b^{i-1} + n_{i-2} * b^{i-2} + ... + n_1 * b^1 + n_0 * b^0$

Conversão de Binária (0,1) para Decimal utilizando o Polinômio Geral

$(101101)_2 = 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$
=32+0+8+4+0+1
=$(45)_{10}$

Por divisões sucessivas encontre os seguintes valores abaixo, lembrando que o restos devem ser **sempre menores** que a base em questão e a montagem dos números seguem de baixo para cima.

Exercícios:

$(46)_{10} = (?)_2$	$(123)_{10} = (?)_2$	$(4305)_{10} = (?)_2$	$(146)_{10} = (?)_2$
$(309)_{10} = (?)_2$	$(1010111)_2 = (?)_5$	$(210011)_3 = (?)_5$	$(376)_{10} = (?)_7$
$(9450)_{10} = (?)_9$	$(1101011)_2 = (?)_4$	$(452)_8 = (?)_2$	$(13215)_6 = (?)_5$

1.1.2 Números Reais

$(123,456)_{10} = 1 * 10^2 + 2 * 10^1 + 3 * 10^0 + 4 * 2^{-1} + 5 * 10^{-2} + 6 * 10^{-3}$

$(123,45)_{10} = (?)_2$

1ª Etapa:

123/2=1111011

2ª Etapa:

$0,45*2=\mathbf{0},90 \rightarrow 0,90*2=\mathbf{1},80 \rightarrow 0,80*2=\mathbf{1},60 \rightarrow 0,60*2=\mathbf{1},20$

$\rightarrow 0,20*2=\mathbf{0},40 \rightarrow 0,80*2=\mathbf{1},60$

$(1111011,01\overline{1100})_2$

Ex:

$(101101,11101)_2 = (?)_{10} = 45,90625$

Operações:

Adição:

$$\begin{array}{ccc} 1 & 1111 & 2 \\ (121)_{10} & (1011011)_2 & (1232)_5 \\ +(39)_{10} & +(11110)_2 & +(32)_5 \\ \hline (160)_{10} & (111001)_2 & (1444)_5 \end{array}$$

Subtração:

$$\begin{array}{ccc} & 111 & 2 \\ (121)_{10} & (1011011)_2 & (1232)_5 \\ -(39)_{10} & -(11110)_2 & -(32)_5 \\ \hline (82)_{10} & (111001)_2 & (1200)_5 \end{array}$$

1.1.3 Exercícios de Fixação

a)$(10346)_{10}$ =$(?)_2$	b)$(156,23)10$ =$(?)_2$	c)$(305,34)_{10}$=$(?)_2$
d)$(786,46)_{10}$=$(?)_2$	e)$(1001110011)_2$=$(?)_{10}$	f)$(101101,1011)_2$=$(?)_{10}$
g)$(1010,100)_2$=$(?)_{10}$	h)$(1111,111)_2$=$(?)_{10}$	i)$(4305,009)_{10}$=$(?)_2$
j)$(200,002)_{10}$=$(?)_2$	l)$(110011,1100)_2$ =$(?)_{10}$	m)$(10110011,11)_2$=$(?)_{10}$

Somas: (da questão anterior)

a)$(g+h)_2$ =$(?)_2$

b)$(e+f)_{10}$ =$(?)_{10}$

c)$(l+m)_2$ =$(?)_2$

d)$(i+j)_{10}$ =$(?)_{10}$

e)$(a+b)_2$ =$(?)_2$

f)$(c+d)_{10}$ =$(?)_{10}$

Subtrações:(da questão anterior)

a)(a-b)=$(?)_2$

b)(c-d)=$(?)_2$

c)(e-f)=$(?)_2$

Tabela de Conversões de Unidades

Decimal	Binário	Quartenário	Octal	Hexadecimal
0	0	0	0	0
1	1	1	1	1
2	10	2	2	2
3	11	3	3	3
4	100	10	4	4
5	101	11	5	5
6	110	12	6	6
7	111	13	7	7
8	1000	20	10	8
9	1001	21	11	9
10	1010	22	12	A
11	1011	23	13	B
12	1100	30	14	C
13	1101	31	15	D
14	1110	32	16	E
15	1111	33	17	F

Exercícios:

$(46)_4 = (?)_2$ $(123)_4 = (?)_2$ $(4305)_4 = (?)_2$ $(146)_4 = (?)_2$

$(307)_8 = (?)_2$ $(4531)_8 = (?)_2$ $(1074)_8 = (?)_2$ $(5076)_8 = (?)_2$

$(9450)_{16} = (?)_2$ $(1AFDC)_{16} = (?)_2$ $(FEDCBA)_{16} = (?)_2$ $(DB452)_{16} = (?)_2$

$(A51F)_{16} = (?)_8$ $(DBA4)_{16} = (?)_8$ $(2100,11)_{16} = (?)_8$ $(376,8)_{16} = (?)8$

$(9450)_{16} = (?)_4$ $(E21A)_{16} = (?)_4$ $(E94,50)_{16} = (?)_4$ $(B45,F)_{16} = (?)_4$

$(1023)_4 = (?)_{16}$ $(765432)_8 = (?)_{16}$ $(65,42)_8 = (?)_{16}$ $(45,7)_8 = (?)_{16}$

$(309)_8 = (?)_4$ $(74777)_8 = (?)_4$ $(76,72)_8 = (?)_4$ $(37,6)_8 = (?)_4$

Portas Lógicas

1.2 Portas Lógicas - Definição

As portas lógicas são circuitos eletrônicos destinados a executar as **Operações Lógicas**. Estes circuitos eletrônicos, compostos de transistores, diodos,resistores, etc, são encapsulados na forma de **Circuito Integrado**.Cada circuito integrado pode conter várias Portas Lógicas, de iguais ou diferentes **Funções Lógicas**.

Portas lógicas de mesma função podem ter características elétricas diferentes, como: corrente de operação, consumo e velocidade de transmissão. Os circuitos integrados, serão estudados os aspectos referentes somente à lógica. Para a eletrônica digital, os símbolos "0"e "1"da álgebra booleana, são níveis de tensão elétrica, onde "0"− Equivale ao nível de tensão mais baixo e "1"− Equivale ao nível de tensão mais alto. Estes níveis lógicos serão os estados lógicos das variáveis lógicas de entrada esaída dos circuitos lógicos.

1.2.1 Tipos de portas lógicas

A seguir serão apresentados os tipos de portas lógicas de duas entradas, com símbolo,função,tabela verdade e um Circuito Integrado equivalente comercial. Algumas portas lógicas podem possuir mais de duas entradas e alguns circuitos integrados,podem possuir tipos diferentes de portas lógicas no mesmo encapsulamento.

Conhecida como álgebra de chaveamento, binária, aplicação direta na eletrônica digital.

1.2.2 Tipos de Portas Lógicas

Porta OU (OR)

Representação Algébrica: F = A + B

Ler-se: A função F é equivalente a variável "A"**ou** "B"

Tabela Verdade

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

Diagrama de Blocos

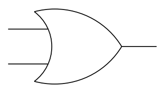


Figura 1.1: Porta OU de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	0	1
B	1	1

Tabela Verdade

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Diagrama de Blocos

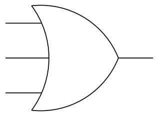


Figura 1.2: Porta OU de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	0	1	1	1
C	1	1	1	1
	$\overline{B}$	B		$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Diagrama de Blocos

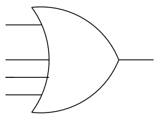


Figura 1.3: Porta OU de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	0	1	1	1	$\overline{D}$
	1	1	1	1	D
C	1	1	1	1	
	1	1	1	1	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

Porta E (AND)

Representação Algébrica: F = A * B

Ler-se: A função F é equivalente a variável "A"**e** "B"

Tabela Verdade

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Diagrama de Blocos

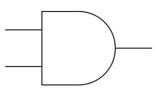


Figura 1.4: Porta E de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	0	0
B	0	1

Tabela Verdade

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Diagrama de Blocos

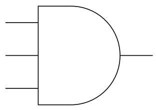


Figura 1.5: Porta E de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	0	0	0	0
C	0	0	0	1
	$\overline{B}$	B	B	$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Diagrama de Blocos

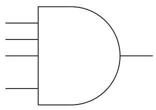


Figura 1.6: Porta E de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	0	0	0	0	$\overline{D}$
$\overline{C}$	0	0	0	0	D
C	0	0	1	0	D
C	0	0	0	0	$\overline{D}$
	$\overline{B}$	B	B	$\overline{B}$	

Porta Inversora (NOT)

Representação Algébrica: F = $\overline{A}$

Ler-se: A função F é equivalente a variável **não** "A"

Tabela Verdade

A	F
0	1
1	0

Diagrama de Blocos

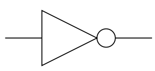


Figura 1.7: Porta Inversora.

Mapa de Karnaugh

$\overline{A}$	A
1	0

Porta Não OU (NOR)

Representação Algébrica: F = $\overline{A + B}$

Ler-se: A função F **não** é equivalente a variável "A"ou "B"

Tabela Verdade

A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

Diagrama de Blocos

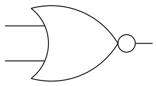


Figura 1.8: Porta Não OU de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	1	0
B	0	0

Tabela Verdade

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Diagrama de Blocos

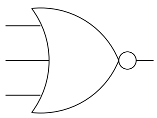


Figura 1.9: Porta Não OU de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	1	0	0	0
C	0	0	0	0
	$\overline{B}$	B		$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Diagrama de Blocos

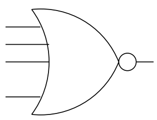


Figura 1.10: Porta Não OU de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	1	0	0	0	$\overline{D}$
	0	0	0	0	D
C	0	0	0	0	
	0	0	0	0	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

Porta Não E (NAND)

Representação Algébrica: F = $\overline{A * B}$

Ler-se: A função F **Não** é equivalente a variável "A"**e** "B"

Tabela Verdade

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

Diagrama de Blocos

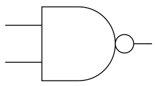


Figura 1.11: Porta Não E de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	1	1
B	1	0

Tabela Verdade

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Diagrama de Blocos

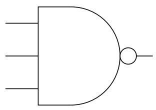


Figura 1.12: Porta Não E de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	1	1	1	1
C	1	1	1	0
	$\overline{B}$	B		$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Diagrama de Blocos

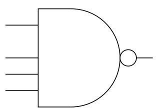


Figura 1.13: Porta Não E de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	1	1	1	1	$\overline{D}$
	1	1	1	1	D
C	1	1	0	1	
	1	1	1	1	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

Porta OU Exclusivo (XOR)

Representação Algébrica: F = $(\overline{A} * B)$+$(A * \overline{B})$ ou A (+) B

Ler-se: A função F é equivalente **ou** a variável "A"**ou** "B"

Tabela Verdade

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Diagrama de Blocos

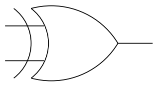


Figura 1.14: Porta OU Exclusivo de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	0	1
B	1	0

Tabela Verdade

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Diagrama de Blocos

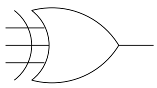


Figura 1.15: Porta OU Exclusivo de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	0	1	0	1
C	1	0	0	0
	$\overline{B}$	B		$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Diagrama de Blocos

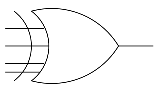


Figura 1.16: Porta OU Exclusivo de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	0	1	0	1	$\overline{D}$
	1	0	0	0	D
C	0	0	0	0	
	1	0	0	0	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

Porta Não OU Exclusivo (XNOR)

Representação Algébrica: F = $(\overline{A}+B)$*$(A+\overline{B})$ ou A (*) B

Ler-se: A função F **não** é equivalente **ou** a variável "A"**ou** "B"

Tabela Verdade

A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Diagrama de Blocos

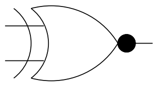


Figura 1.17: Porta Não OU Exclusivo de 2 entradas.

Mapa de Karnaugh

	$\overline{A}$	A
$\overline{B}$	1	0
B	0	1

Tabela Verdade

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Diagrama de Blocos

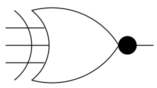


Figura 1.18: Porta Não OU Exclusivo de 3 entradas.

Mapa de Karnaugh

	$\overline{A}$		A	
$\overline{C}$	1	0	1	0
C	0	1	1	1
	$\overline{B}$	B	B	$\overline{B}$

Tabela Verdade

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Diagrama de Blocos

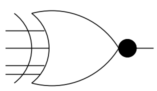


Figura 1.19: Porta Não OU Exclusivo de 4 entradas.

Mapa de Karnaugh

	$\overline{A}$		A		
$\overline{C}$	1	0	1	0	$\overline{D}$
$\overline{C}$	0	1	1	1	D
C	1	1	1	1	D
C	0	1	1	1	$\overline{D}$
	$\overline{B}$	B	B	$\overline{B}$	

1.2.3 Teoremas

Teoremas de D'Morgam ou Morgan

1ª **Teorema**

$$\overline{A} + \overline{B} = \overline{A * B}$$

2ª **Teorema**

$$\overline{A} * \overline{B} = \overline{A + B}$$

Demonstração

1° **Teorema**

A	B	1° Mem	2° Mem
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

2° **Teorema**

A	B	1° Mem	2° Mem
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Principais Postulados de Boole

Considere X, Y e Z variáveis lógicas distintas.

0 * X = 0

1 * X = X

X * X = X

$\overline{X}$ * X = 0

0 + X = X

1 + X = 1

X + X = X

$\overline{X}$ + X = 1

$\overline{\overline{X}}$ = X

Comutativas:

X + Y = Y + X

X * Y = Y * X

Associativas:

X + (Y + Z) = (X + Y) + Z

X * (Y * Z) = (X * Y) * Z

Distributivas:

X * (Y + Z) = (X * Y) + (X * Z)

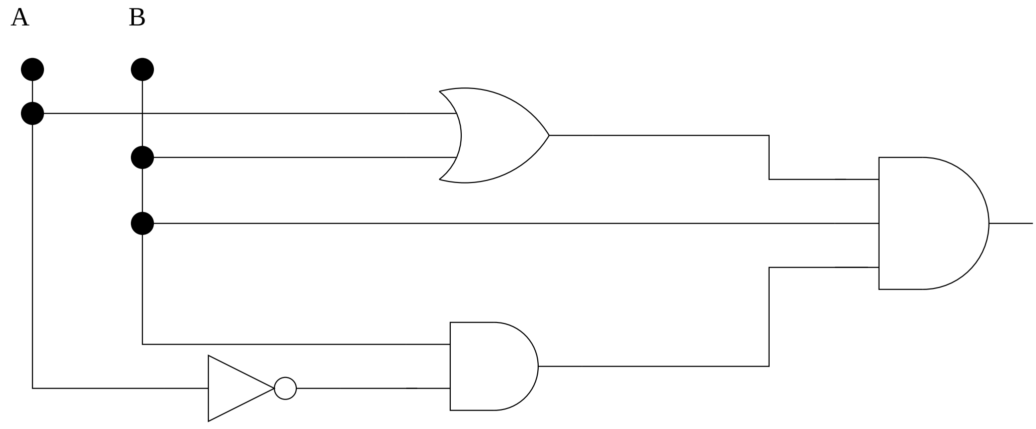
1.2.4 Exercícios:

Dado a função abaixo, monte a tabela verdade, o mapa de Karnaugh e o Diagrama de Blocos.

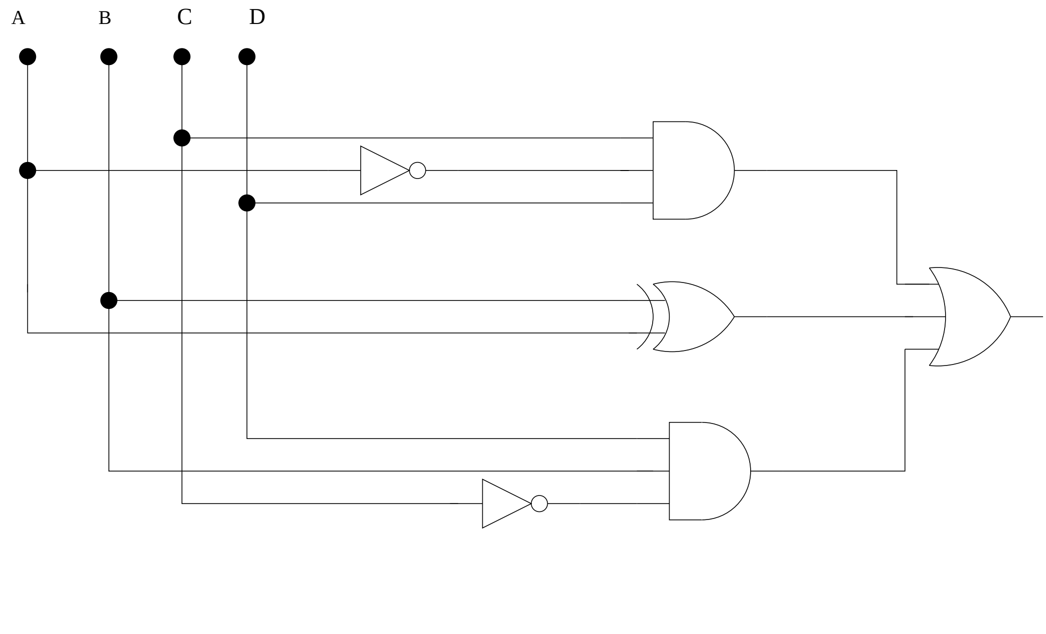
a)F=(A+B) * $\overline{C}$

b)F= A * $\overline{B}$ + A*B*$\overline{C}$ +A*C

c)Monte a expressão e simplifique-a



d)Monte a expressão e simplifique-a



e)Monte a expressão e o diagrama de blocos

	$\overline{A}$		A	
$\overline{C}$	X	0	1	X
C	0	1	0	0

f)Monte a expressão e o diagrama de blocos

	$\overline{A}$		A		
$\overline{C}$	0	1	0	0	$\overline{D}$
	0	1	1	1	D
C	1	1	1	0	D
	0	0	0	0	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

g)Monte a expressão e o diagrama de blocos

	$\overline{A}$		A		
$\overline{C}$	X	0	X	1	$\overline{D}$
	1	X	0	1	D
C	1	0	0	0	D
	1	1	0	0	$\overline{D}$
	$\overline{B}$	B		$\overline{B}$	

1.3 Exercícios de Fixação:

a)Projetar um sistema para a identificação da altura de garrafas produzidas poruma empresa de cerveja. Sabe-se que a empresa produz garrafas com 3 alturaspadronizadas 10 cm, 15 cm e 20 cm. As garrafas abandonam a linha de produção naposição vertical transportada por uma esteira. Utilizar sensores ópticos eindicadores de led´s coloridos, uma cor para cada altura de garrafa.

b)Um teclado decimal fornece 4 informações binárias indicando qual tecla que foi pressionada. Deseja dimensionar um sistema digital que acenda um led sempre que a tecla pressionada seja múltipla de 2 **ou** de 3.

c)Um teclado decimal apresenta saída codificada em binário. Escrever a equação algébrica simplificada de uma função de chaveamento (lógica) que indique sempre qua a tecla pressionada seja um número impar.

d)Projetar um sistema lógico conversor do código BCD para um display de 7 segmentos.

e)Dimensionar um sistema lógico que recebendo em suas entradas um código BCD mostre em um display de 7 segmentos os seguintes requesitos:
0 → U
Par → L
Impar → A

Simplifique as expressões:

$S = A * \overline{B} * C + \overline{A * C} + \overline{A + B}$

$S = \overline{A * B} * C + A * B * \overline{C} + A * \overline{B + C}$

$S = A * B + \overline{A * B}$

$S = \overline{A * B} + \overline{C} + \overline{A * B * C} + A\overline{B + C} + \overline{A} * B * C + A * \overline{B} * C$

Capítulo 2

Segunda Unidade

2.1 Sistemas Digitais

Um sistema digital e um conjunto de funções de chaveamento envolvendo variáveis binárias e que realizam determinadas tarefas. Os sistemas digitais se agrupam em duas categorias distintas:

a)Sistemas Digitais Combinacionais, e

b)Sistemas Digitais Seqüenciais.

Os sistemas combinacionais apresentam em suas saídas, num certo instante de tempo, valores que dependem exclusivamente dos valores aplicados em suas entradas nesse exato instante.
Os sistemas seqüências apresentam em suas saídas, em um determinado instante,valores que dependem dos valores presentes nas entradas nesse instante e em instantes anteriores.

2.1.1 Flip-Flop-SR

Para tal comportamento os sistemas seqüenciais deverão conter estruturas de memorização que armazenarão entradas anteriormente aplicadas. O modulo básico de memorização são os **FLIP-FLOP**, sendo facilmente construído a partir de portas lógicas introduzindo-se uma realimentação adequada na mesma.

Assim os **FLIP-FLOP** são dispositivos que possuem dois estados estáveis. Para um **FLIP-FLOP** assumir um desses estados e necessário que haja uma combinação das variáveis e de um pulso de controle, clock. Após este pulso, o **FLIP-FLOP** permanecera nesse estado até a chegada de um novo pulso de controle e, então, de acordo com as variáveis de entrada, permanecerá ou mudará de estado.

Basicamente, podemos representar o **FLIP-FLOP** como um bloco onde temos duas saídas Q e $\bar{Q}$, entradas para as variáveis e um entrada de controle (clock). A saída Q será a principal do bloco.

	S	R	Qa/Qn	Qf/Qn+1
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

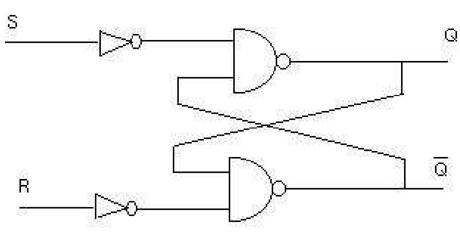


Figura 2.1: Flip-Flop SR discreto.

Onde Qa/Qn representa o estado anterior e Qf ou Qn+1 o estado possível.

Assim podemos assumir que a tabela verdade de um flip-flop SR básico e:

S	R	Qf
0	0	Qa
0	1	0
1	0	1
1	1	Não permitido

Existem vários tipos de FLIP-FLOP classificados em dois grandes blocos:

- Síncrono
- Assíncrono

Os FLIP-FLOP síncronos só respondem as mudanças de estados nas entradas quando essas ocorrem simultaneamente com a ocorrência de um pulso de controle (clock ou triger), ou seja, o sincronismo, enquanto que os assíncronos reagem quanto à variação das entradas.

Além dessas classificações os FLIP-FLOP se agrupam em algumas famílias, ou tipos como:

1. Set-Reset (SR);
2. Master-Slave(MS);
3. JK;
4. Tipo T, e;
5. Tipo D (Delay)

2.1.2 Flip-Flop SR controlado por um pulso de Clock

Para que o flip-flop SR básico seja controlado por uma seqüência de pulsos de clock, basta trocarmos os dois inversores por portas NAND, e as outras entradas destas portas, injetarmos o clock. O circuito ficará, então:

Quando a entrada clock assumir o valor 1, o circuito ira comportar-se como um flip-flop SR básico. Teremos então, a seguinte tabela verdade:

S	R	Qf
0	0	Qa
0	1	0
1	0	1
1	1	Não permitido

Esse circuito ira mudar de estado apenas quando o clock for igual a 1, em outras palavras, o circuito irá mudar de estado somente na chegada de um pulso de clock.

Diagrama de Estados

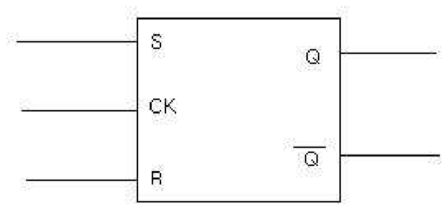


Figura 2.3: Flip-Flop SR Bloco com clock

Figura 2.2: Flip-Flop SR discreto com clock

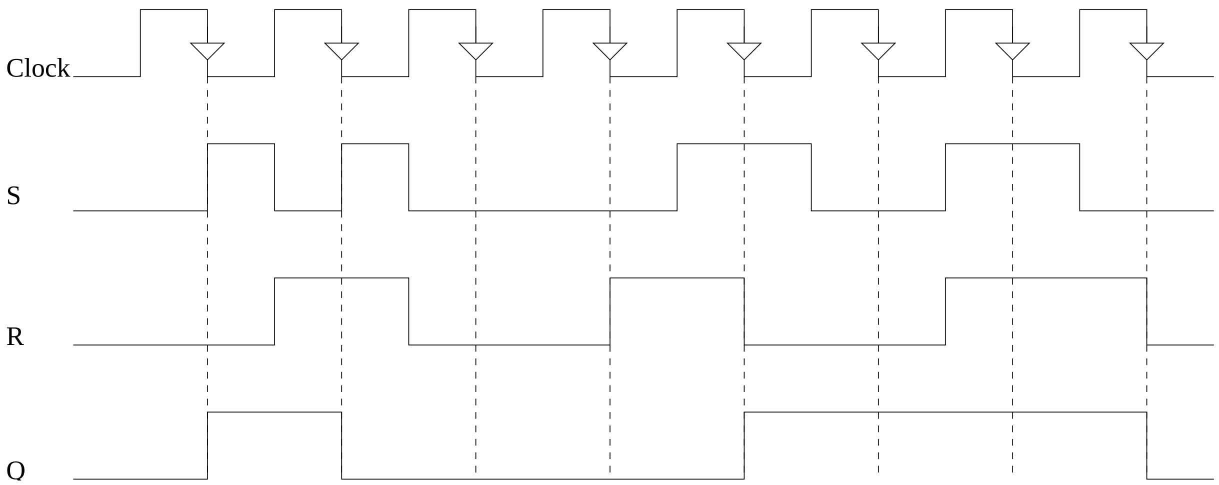


Figura 2.4: Diagrama de Estados do Flip-Flop SR

2.1.3 Flip-Flop JK

O flip-flop JK, nada mais e que um SR realimentado de maneira mostrada na figura a seguir, essa outra forma de realimentação elimina o estado indefinido do flip-flop SR.

A tabela verdade fica:

	J	K	Qa	$\overline{Qa}$	S	R	Qf
0	0	0	0	1			Qa
1	0	0	1	0			Qa
2	0	1	0	1			0
3	0	1	1	0			0
4	1	0	0	1			1
5	1	0	1	0			1
6	1	1	0	1			Qf
7	1	1	1	0			Qf

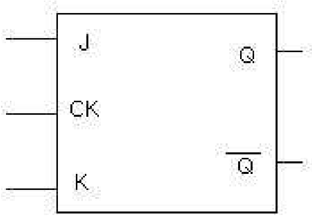


Figura 2.6: Flip-Flop JK Bloco

Figura 2.5: Flip-Flop JK discreto

OBS: Vale ressaltar para que o circuito assim funcione como desejado, deve-se retirar o clock logo após as duas entradas tenham sido iguais a 1.

2.1.4 Flip-Flop JK com entradas Preset e Clear

O Flip-Flop JK poderá assumir valores $Q = 1$ ou $Q = 0$ mediante a utilização das entradas Preset (Pr) e Clear (Clr). Estas entradas são inseridas no circuito da seguinte forma:

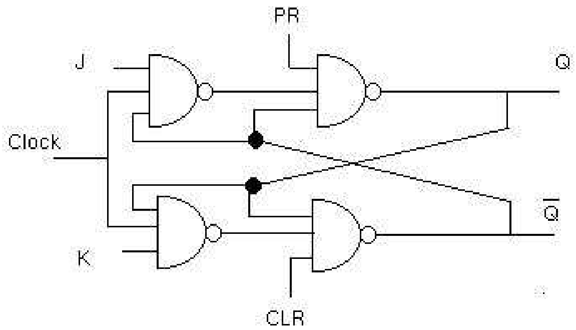


Figura 2.7: Flip-Flop JK com Preset Clear

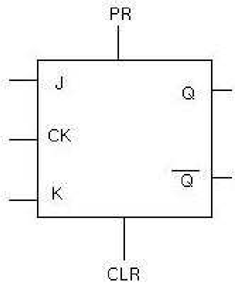


Figura 2.8: Flip-Flop JK com Preset Clear

As entradas Preset e Clear não podem assumir valores zero simultaneamente, pois acarretaria a saída uma situação não permitida. A entrada Clear e também denominada de **Reset**.

CLR	PR	Qf
0	0	Não permitido
0	1	0
1	0	1
1	1	Funcionamento Normal

2.1.5 Flip-Flop JK Master-Slave (Mestre-Escravo)

O flip-flop JK como foi visto, resolveu o problema anteriormente visto, quando as entradas J e K forem iguais a 1 porem, este circuito apresenta uma característica indesejável, quando o clock for igual a 1, teremos o circuito funcionando como um sistema combinacional, pois a entrada J e K estarão liberadas. Para solucionarmos o problema utilizaremos o circuito abaixo:

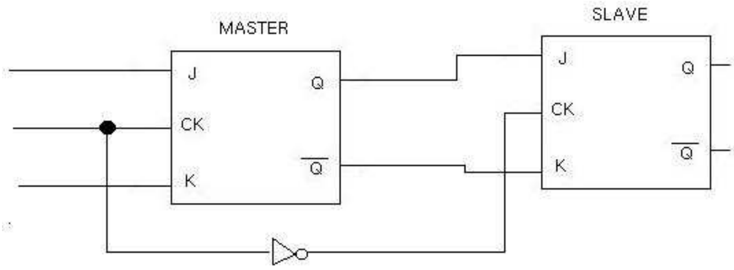


Figura 2.9: Flip-Flop JK Master-Slave

2.1.6 Flip-Flop T

Esse e um flip-flop JK com a particularidade de possuir as entradas J e K curto circuitadas (uma ligada a outra), logo quando J assumir valor 1, K também assumira o valor 1, e quando J assumir valor zero, K também.

Figura 2.10: Flip-Flop T

2.1.7 Flip-Flop D

Esse e um flip-flop JK com a particularidade de possuir as entradas J e K invertidas. Logo, nesse flip-flop, teremos as seguintes entradas possíveis: J=0 e K=1; J=1 e K=0.

Ex1 :Projetar um sistema bloqueador de bêbados num carro. A seqüência da senha devera ser 101

Ex2 :Projetar um sistema seqüencial síncrono que simule um dado eletrônico. Utilizar flip-flop JK.

Figura 2.11: Flip-Flop D

Ex3 :Projetar um sistema seqüencial síncrono usando flip-flop JK que acionado por um gerador de clock em um display de 7 segmentos de forma seqüencial e cíclico, as letras que compõem o nome: LEAO.

Registradores

2.2 Registradores de Deslocamento

Os flip-flop podem armazenar durante o período em que sua entrada de clock for igual a 0, um bit apenas (saída Q). Porem quando necessitarmos guardar um informação de mais de um bit, o flip-flop ira tornar-se insuficiente. Contornar tal problema costuma-se utilizar no circuito o que se denomina Registradores de Deslocamento (*Shift Register*). Assim com um certo número de flip-flop do tipo RS ou JK mestre-escravo ligados de tal forma que as saídas de cada bloco alimentem as entradas S e R, respectivamente, do flip-flop seguinte, sendo que, o primeiro terá suas entradas S e R ligadas na forma de um flip-flop tipo D (R=S). O circuito abaixo exemplifica um Registrador de Deslocamento.

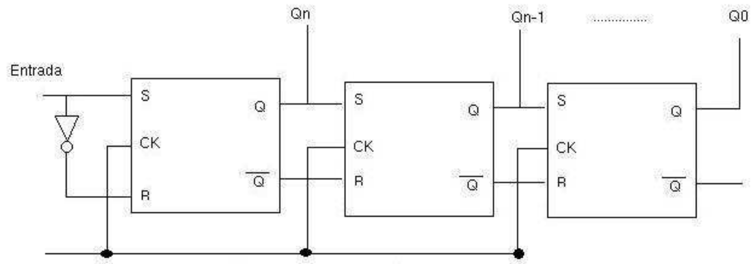


Figura 2.12: Registrador de Deslocamento Simples

Veremos então algumas aplicações do registrador de deslocamento.

2.2.1 Conversor Série-Paralelo

O Registrador de deslocamento pode ser utilizado para converter uma informação série em paralela. A configuração básica, nessa situação, para uma informação de 4 bits, teremos:

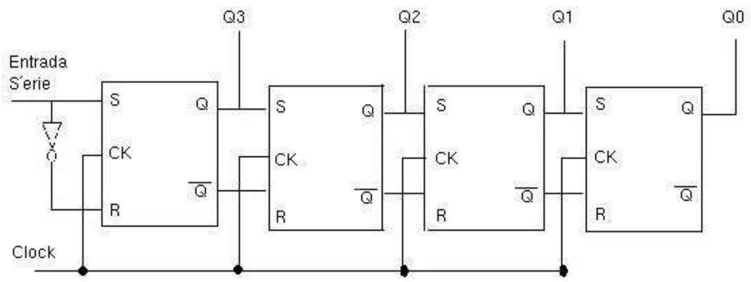


Figura 2.13: Conversor Série - Paralelo

Fazendo a seguinte entrada serie 1010 no circuito acima teremos a tabela verdade da seguinte forma:

Informação	Descidas do Clock	Q3	Q2	Q1	Q0
0	1 Pulso	0	0	0	0
1	2 Pulso				
0	3 Pulso				
1	4 Pulso				

Por esse motivo o circuito acima e conhecido como Registrador de Deslocamento.

2.2.2 Conversor Paralelo - Série

Para entrarmos com uma informação paralela, necessitamos de um registrador que apresente as entradas Preset e Clear, pois e através destas que fazemos com que o Registrador armazene a informação paralela. O registrador com essas entradas e representado abaixo:

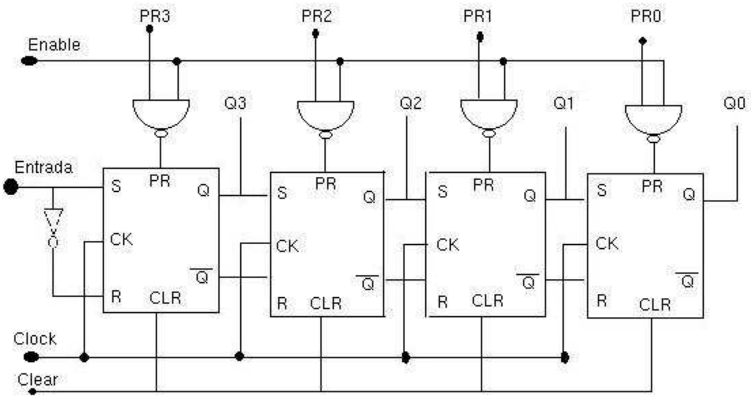


Figura 2.14: Conversor Paralelo - Série

Antes de começarmos, vamos rever o funcionamento das entradas ENABLE e PRESET. Quando a entrada enable estiver em zero, as entradas preset (PR) dos flip-flop permanecerão no estado 1, fazendo com que os flip-flop atuem normalmente. Quando a entrada enable for igual a 1, as entradas preset dos flip-flop assumirão os valores complementares das entradas PR3, PR2, PR1 e PR0.

Para que o registrador de deslocamento funcione como conversor paralelo série, necessitamos limpá-lo e logo em seguida, introduzir a informação como já descrito, recolhendo na saída Q0 a mesma informação de modo serie. E fácil de notar que a saída Q0 assume primeiramente o valor I0 e a cada descida do pulso de clock, ira assumir seqüencialmente os valores I1, I2, I3.

Informação	Descidas do Clock	Q3	Q2	Q1	Q0
0	1 Pulso	0	0	0	0
1	2 Pulso				
0	3 Pulso				
1	4 Pulso				

Contadores

2.3 Contadores

São sistemas seqüenciais que contam o numero de pulsos que ocorre em sua entrada durante um certo intervalo de tempo. A indicação da contagem e dada na base 2 e obtida através das saídas binárias do contador. Existem dois tipos básicos de contadores:

a)Os Assíncronos - dos quais as transições dos Flip-Flop não são simultâneos.

b)Os Síncronos - dos quais as transições dos Flip-Flop são simultâneas e geradas por um sinal de clock.

2.3.1 Contadores Assíncronos

São caracterizados por não terem entradas de clocks comuns. Essa se faz apenas no 1 flip-flop e as outras entradas de clock dos outros flip-flop serão funções das saída. Os contadores assíncronos podem ter módulos binário e módulos não binário.

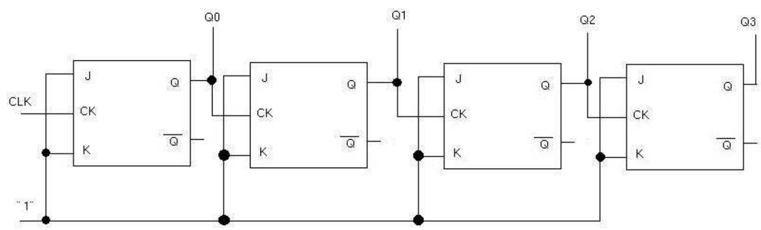


Figura 2.15: Contador Assíncrono

A principal característica de um contador de pulso e representar o código BCD 8421. Seu circuito básico apresenta um grupo básico de 4 flip-flop JK mestre-escravo os quais possui as entradas J=K=1.

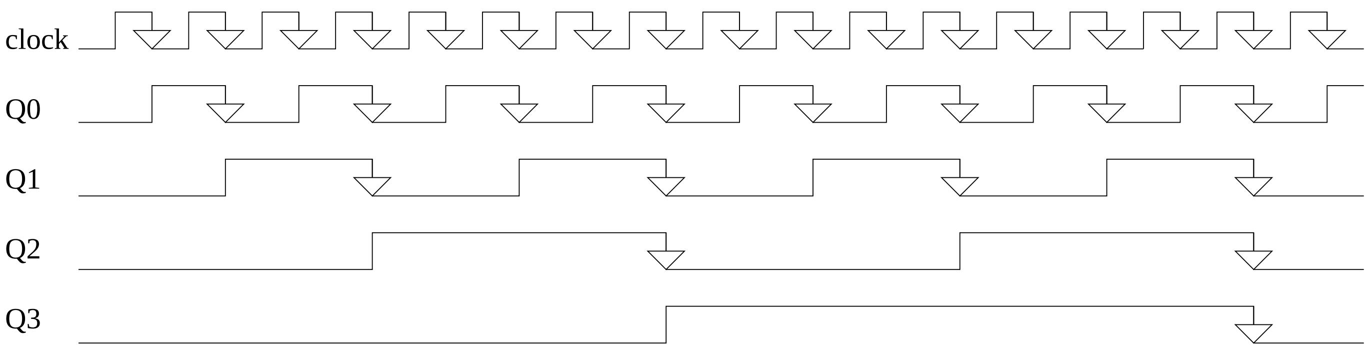


Figura 2.16: Diagrama de estado

2.3.2 Contadores Síncronos

Neste tipo de contador todos os flip-flop são liberados na mesmo instante, pois estes contadores possuem as entradas de clock curto-circuitadas, ou seja, o clock aciona todos os flip-flop simultaneamente. A indicação da contagem pode ser obtida diretamente das saídas dos flip-flop ou através de circuitos combinacionais. O numero de flip-flop necessários para cada contador depende do modulo do contador apartar da seguinte expressão: $2^{n-1} \leq M \leq 2^{n}$, onde n e o numero de flip-flop. Para estudarmos os contadores síncronos devemos sempre escrever a tabela verdade, estudando assim quais devem ser as entradas J e K dos vários flip-flop e que estes assumam o estagio seguinte.

Para isso devemos lembrar então da tabela verdade do JK.

	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

Ex: Utilizando flip-flop JK com Preset-Clear projetar um contador cíclico para a seqüência abaixo:

$$\begin{array}{ccccc} 0 & \rightarrow & 1 & \rightarrow & 2 \\ \uparrow & & & & \downarrow \\ 5 & \leftarrow & 4 & \leftarrow & 3 \end{array}$$

Sistema de Projetos

2.4 Sistema de Projetos de Subsistemas Seqüenciais

O projeto de subsistemas (pequenos sistemas básicos) seqüenciais seguem os seguintes passos:

a) A partir da descrição verbal do sistema deve-se construir um diagrama de estados no qual são identificados os vários estados distintos que o sistema apresenta, as transições que devem ocorrer entre esses estados, assim como as saídas que devem ser produzidas.

b) Os diferentes estados identificados deverão ser designados(identificados)pelas combinações das saídas dos flip-flop utilizados no sistema.

c) As transições entre estados desejados serão produzidas pela aplicação adequada de variáveis da excitação nas entradas do flip-flop de modo a produzir as mudanças adequadas. Essas variáveis serão criadas a partir das variáveis de estado (saída dos flip-flop).

d) As variáveis de saída deverão ser criadas a partir das variáveis de estado de acordo com a descrição do sistema.

Os sistemas seqüenciais poderão ser síncronos quando todos os flip-flop receberem o mesmo clock, enquanto o sistema reagir apenas aos sinais presentes na entrada simultaneamente com o clock, ou serão assíncronos quando o sistema reagir aos sinais de entrada no instante que esses forem aplicados, neste caso não existira um clock único para os flip-flop.

	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Ex: Dimensionar um sistema seqüencial síncrono que recebendo em sua entrada 2 informações binárias X e Y (sincronizadas com o clock), produz uma saída única Z, sempre que pela terceira vez consecutiva as 2 entradas, X e Y forem iguais. Toda vez que o sistema produzir uma saída Z=1 devera se rearmar para iniciar uma nova codificação.

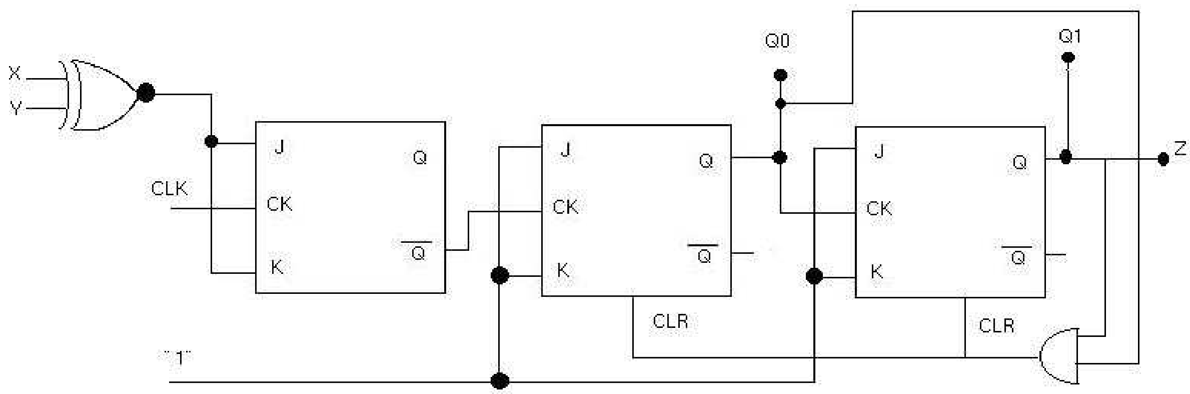


Figura 2.17: Uma das possíveis resolução do exercício

Capítulo 3

Terceira Unidade

3.1 Conversores A/D e D/A

3.1.1 Introdução

A maioria dos dados obtidos de sensores comuns, tais como sensores de temperatura, intensidade luminosa, posição, tensão, corrente e etc. fornecem sinais analógicos, ou seja, uma tensão que é proporcional à grandeza medida e que varia de forma contínua numa faixa de valores.

No entanto, a maioria dos equipamentos modernos que fazem a aquisição de dados destes sensores, trabalha com técnicas digitais. Isso significa que o dado analógico, preciso ser convertido para a forma digital. Para fazer esta conversão são utilizados circuitos denominados conversores analógico-digital, ou simplesmente A/D, como seu próprio nome indica, realiza a conversão de sinais, cuja amplitude varia continuamente em sinais digitais correspondentes à amplitude do sinal original.

Para converter se faz o uso de um comparador de tensão ou corrente - variando de acordo com a aplicação - que irá comparar o sinal analógico com o valor de referência.

Desta forma os circuitos A/D devem preencher certos requisitos importantes quanto ao seu desempenho que são:

- Quantização;
- Taxa de Amostragem e;
- Linearidade.

3.1.2 Quantização

Entre os dois valores extremos da escala de valores analógicos que devem ser convertidos para a forma digital existem infinitos valores intermediários, o que justamente caracteriza uma grandeza que varia de forma análoga ou analógica.
Entretanto, quando passamos um valor qualquer entre os dois valores extremos incluindo-os, não podemos representar qualquer quantidade, pois precisaríamos para isso de um número infinito de bits.

Assim, por exemplo, se utilizarmos na conversão 4 bits, teremos a possibilidade de representar apenas 16 valores na escala total de valores analógicos, e se usarmos 8 bits poderemos representar 256 valores, conforme indica a figura 3.1.
Se tivermos uma escala de 0 a 8 V, por exemplo, e usarmos 4 bits para a conversão, os "degraus"da escada de conversão terão 0,5 V de altura, o que significa que este conversor terá uma resolução de 0,5 V. Se usarmos um conversor A/D de 8 bits (256 "degraus"de resolução) para fazer um voltímetro de 0 a 10 V por exemplo, a resolução deste voltímetro será de 10/256 ou pouco menos de 0,04 V.

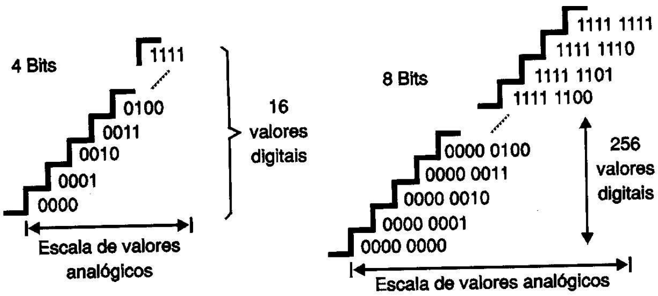


Figura 3.1: Escala de conversão

Este comportamento "digital"pode ser observado em muitos instrumentos comuns, tais como os multímetros digitais em que, se a grandeza medida estiver num valor intermediário entre dois degraus da resolução do conversor A/D, o valor apresentado no display oscilará entre eles.

Evidentemente, tanto maior é a precisão na conversão mais bits serão utilizados pelo conversor. Tipos com 8 a 16 bits são comuns nas aplicações industriais e em medidas, dependendo da quantidade de "passos"desejados na conversão ou a resolução.

3.1.3 Taxa de Amostragem

Muitos processos de aquisição de dados de sensores, de processos ou de outras aplicações precisam ser rápidos. Uma placa de aquisição de dados de um instrumento de medida que projete uma forma de onda, desenhe um gráfico na tela de um PC representando um processo dinâmico ou mesmo um instrumento digital simples como um multímetro, devem estar constantemente em andamento.

Um osciloscópio digital, por exemplo, deve medir as tensões instantâneas de um sinal em diversos pontos ao longo de um ciclo para poder "desenhar"esta forma de onda com precisão na tela. Se a freqüência do sinal for alta, isso implica a necessidade de se fazer amostragens num tempo extremamente curto.

Os conversores A/D podem ser encontrados em tipos que têm freqüências de amostragem numa ampla escala de valores. Os tipos mais rápidos têm suas velocidades especificadas em MSPS (Mega Samples Per Second ou Mega Amostragens Por Segundo).

Uma máquina industrial ou um instrumento de uso geral como um multímetro pode usar conversores A/D relativamente lentos com taxas ou velocidades de amostragens de até algumas unidades por segundo. Um multímetro digital comum, por exemplo, faz de 1 a 10 amostragens por segundo apenas, dependendo do tipo. Todavia, um osciloscópio digital ou virtual que precise observar uma forma de onda de 10 MHz, deve, para ter uma definição razoável, realizar pelo menos 100 milhões de amostragens por segundo (10 pontos por ciclo).

3.1.4 Linearidade

A curva de conversão da grandeza analógica para a forma digital deve ser linear para um bom conversor. Isso significa que não existem desvios na correspondência entre o valor analógico e a saída digital ao longo da escala de valores em que o conversor deve trabalhar.

No entanto, na prática podem ocorrer pequenos desvios, de acordo com o que mostra a figura 3.2.

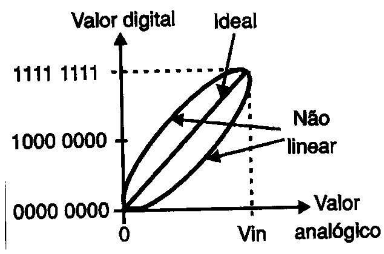


Figura 3.2: Grau de linearidade da conversão

Isso quer dizer que, em determinadas faixas de valores, a conversão pode ser menos precisa. Esta imprecisão é mais grave nos tipos de maior definição, pois os desvios podem ter a mesma ordem de grandeza que os "degraus"da escada de conversão, afetando assim a precisão final da mesma.

3.2 Desenvolvimento

Para fazer uma conversão de sinais analógicos para a forma digital existem diversas técnicas que são empregadas nos circuitos comerciais, muitas delas encontradas em circuitos integrados que são "embutidos"(embedded) em aplicações mais complexas, os quais fazem o controle de máquinas e equipamentos.

Analisamos as tecnologias mais empregadas para esta finalidade começando com o bloco comum a todos os conversores, que é o circuito de amostragem e manutenção (sample and hold).

O valor dos sinais analógicos que devem ser convertidos para a forma digital corresponde a um determinado instante, cuja duração, em alguns casos, não vai além de alguns milionésimos de segundo.

Assim, um primeiro bloco importante do conversor é um circuito que lê o valor do sinal a ser convertido num determinado instante e o armazena de modo que, mesmo que o sinal varie depois, os circuitos que fazem a conversão têm numa memória seu valor. Este circuito é ilustrado em blocos na figura 3.3.

O sinal a ser amostrado é amplificado por um buffer de entrada cuja finalidade é não carregar o circuito externo, e ao mesmo tempo proporcionar isolamento do circuito de conversão.

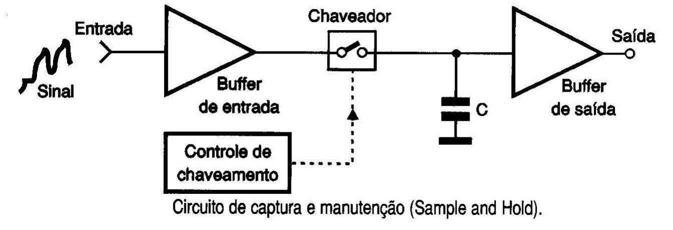


Figura 3.3: Diagrama de blocos de um conversor A/D

Na saída deste circuito temos uma chave eletrônica ou chaveador, que determina o instante exato em que a leitura do sinal deve ser feita. A chave fecha então por uma fração de segundo (numa freqüência que depende da velocidade de amostragem) permitindo que o sinal carregue o capacitor C.

Assim, quando a chave abre, esperando a leitura seguinte, o capacitor tem armazenado o valor da grandeza analógica a ser convertida. Esta tensão no capacitor é mantida no circuito conversor através de um buffer de saída durante o tempo que ele necessita para isso.

Na figura 4 temos um gráfico que indica de que modo à tensão de entrada varia e o circuito de amostragem e retenção mantém a saída constante durante os intervalos de conversão (que correspondem aos "degraus").

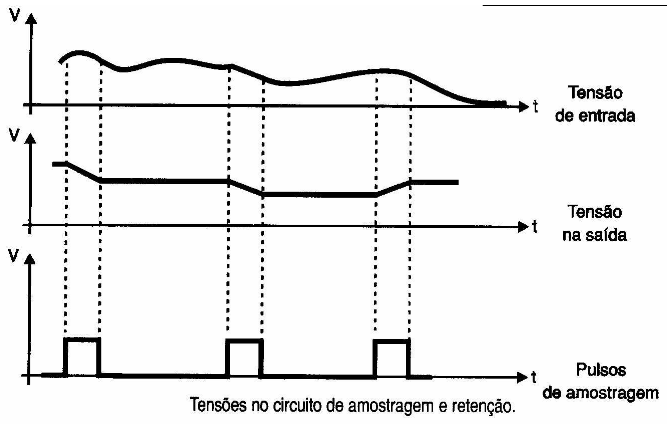


Figura 3.4: Escala de conversão

3.2.1 Aplicação

Desenvolvendo um pequeno programa no Matlab 6.0 podemos exemplificarmos melhor toda esta teoria aqui mostrada. A onda fundamental tem uma freqüência de 120 Hz e está defasada em 60°, atribuímos valores de quantização de: 4, 8 e 12 Bits e taxa de amostragem de: 240, 600 e 1000 Hz (respeitando a freqüência de Nyquist).

Primeiramente o nosso programa vai marcar os tempos que serão armazenados com seus respectivos valores analógicos para posteriormente serem quantizados e assim aplicando a transforma discreta de Fourier reconstituir o sinal amostrado.

Nos gráficos abaixo, podemos verificar que em se tratando de um sinal digital, não existe valores negativos na quantização, o que pode ocorrer que vemos em multímetros digitais ou outros aparelhos são um bit a mais inserido posteriormente a quantização para sinalização se aquele valor se trata de um valor negativo ou positivo, o que não interfere em nada na conversão, com mencionei é apenas uma sinalização para o usuário.

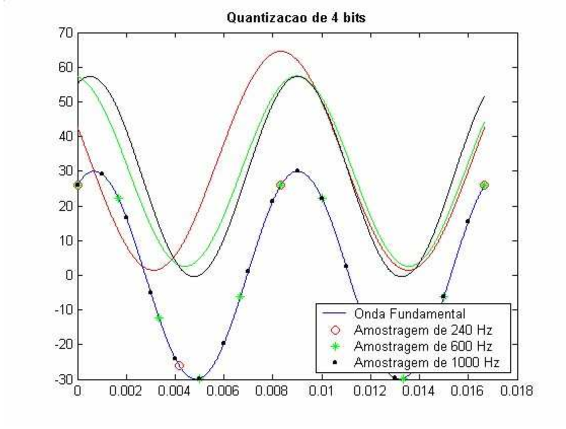


Figura 3.5: Quantização em 4 bits de resolução

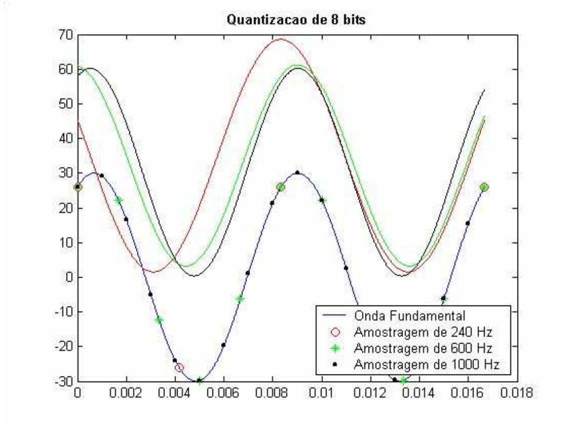


Figura 3.6: Quantização em 8 bits de resolução

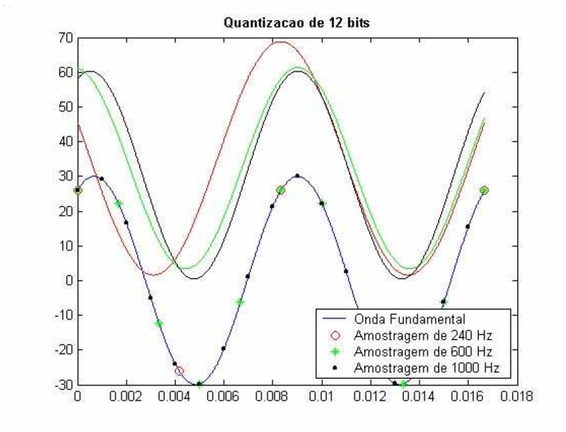


Figura 3.7: Quantização em 12 bits de resolução

Existem várias formas de se construir conversores A/D, sendo que cada um tem a sua característica de funcionamento que deve ser levada em conta, na hora de se construir e/ou escolher para a sua aplicação. Temos uma relação de possíveis combinações:

- Conversor A/D com comparador em paralelo;
- Conversor A/D com rampa em escada;
- Conversor A/D de aproximações sucessivas;
- Conversor A/D de rampa única;
- Conversor A/D de rampa dupla e;
- Sigma-Delta.

O Sigma-Delta é um das importantes técnicas de conversão A/D, utilizada em que se deseja uma altíssima velocidade de conversão, como nos DSPs (Digital Signal Processing).

Portanto, vimos que a conversão do sinal analógico para o digital sempre existe uma perda de informação seja ela de amplitude - característica da quantidade de bits utilizados - ou de fase do sinal - característica da taxa de amostragem empregada.

Vimos que o erro máximo que pode ocorrer na quantização é de metade do valor de nível da quantização assim sendo quanto maior for o número de bits do conversor menor será o seu erro.

O erro de "Aliasing" é facilmente evitado utilizando o teorema da amostragem que "Para que uma determinada freqüência f1 do sinal analógico seja ou possa ser completamente reconstituída a taxa amostral, no processo de digitalização, deve ser no mínimo igual a 2*f1"

Conhecidas as imperfeições da conversão podemos então saber quais os fatores que influem na escolha de um conversor A/D e assim prever melhor os ajustes que sistema deverá sofrer, pois já é sabido as suas fraquezas.

Multiplexadores e Demultiplexadores

3.3 Multiplexadores

No nosso dia a dia lidamos com vários sistemas que utilizam multiplexadores e demultiplexadores, o mais comum deles e o aparelho de som de nossa residência, em uma chave seletora, selecionamos qual fonte sonora a qual utilizaremos (Vinil, CD, Tape, Radio, MD, etc.). A chave seletora então especifica qual o canal de comunicação que será utilizado, conhecida também como via de dados, e assim, está informação será amplificada e transmitida para os auto-falantes. Assim de uma maneira geral, o **MUX**, seleciona um entre vários sinais de entrada e o envia para a saída.

Um multiplexador digital ou seletor de dados é um circuito lógico que aceita diversos dados digitais de entrada e seleciona um deles, em um certo instante, para a saída. O roteamento do sinal de entrada desejado para a saída é controlado pelas entradas de **SELEÇÃO** (conhecidas também como ENDEREÇOS).

O multiplexador atua como uma chave digital controlada de várias posições, onde o código digital aplicado nas entradas de SELEÇÃO controla qual será a entrada de dados chaveada para a saída. Por exemplo, a saída será igual a entrada de dados I0 para um determinado código de SELEÇÃO; e assim será igual a I1 para um outro determinado código de SELEÇÃO; e assim por diante. Em outras palavras, um multiplexador seleciona 1 entre N dados de entrada e transmite o dado selecionado para um único canal de saída. Isto é chamado de **multiplexação**.

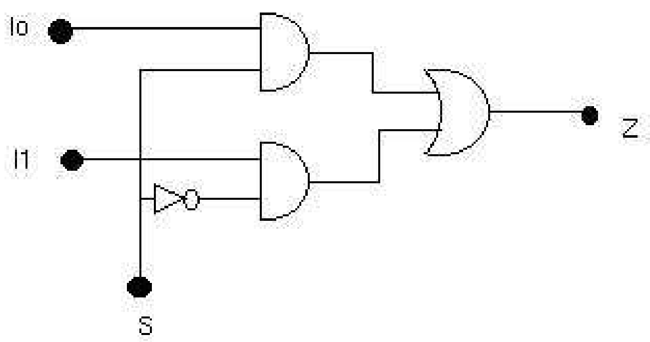


Figura 3.8: Circuito de um multiplexador de 2 entradas

Uma outra aplicação para um multiplexador seria utilizá-lo como um conversor paralelo-série um vez que o seu princípio de funcionamento se adequa a tal finalidade.

3.4 Demultiplexadores

Um multiplexador recebe varias entradas e transmite uma delas para a saída Um demultiplexador (DEMUX) realiza a operação inversa: ele recebe uma única entrada e a distribui por várias saídas. Assim como no multiplexador, o código de SELEÇÃO de entrada determina para qual saída entrada de DADOS será transmitida. Em outras palavras,o demultiplexador recebe uma fonte de dados e seletivamente a distribui para 1 entre N saídas, como se fosse uma chave de varias posições.

As aplicações desses dispositivos são inúmeras desse de sistemas de segurança sistemas complexos de telecomunicações. Para todas as essas aplicações os dois dispositivos devem ser previamente sincronizados para que as entradas serem as mesmas nas saídas.

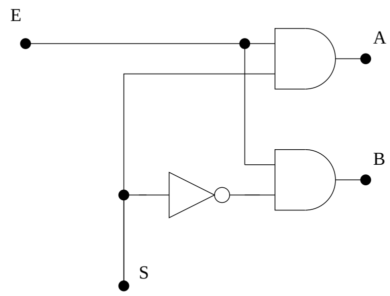


Figura 3.9: Circuito de um demultiplexador de 2 entradas

Circuitos Aritméticos

3.5 Circuitos Aritméticos

Como vimos anteriormente os circuitos combinacionais, vamos encontrar alguns circuitos importantes de grande utilidade e que são a essência da computação hoje existente. São os circuitos aritméticos também muito conhecidos como **ULA** (Unidade Logica Aritmetica).

3.5.1 Meio Somador

Como sabemos, os computadores trabalham na forma binária e já é de se esperar que o mesmo faca suas operações na forma binária. Relembrando a soma de dois números binários teremos:

$$\begin{array}{rrrrrrrr} & & & & & & 1 & \\ & 0 & & 1 & & 0 & & 1 \\ + & 0 & + & 0 & + & 1 & + & 1 \\ & - & & - & & - & & - \\ & 0 & & 1 & & 1 & & 10 \end{array}$$

Montando a tabela verdade teremos:

A	B	Saída (S)	Transporte (Ts)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

O diagrama de blocos seria as saídas receptivas a uma porta lógica especifica como para saída S teremos um XOR e para Ts teremos uma AND. Esse circuito denominado Meio Somador e também conhecido como Half-Adder, termo derivado do inglês.

3.5.2 Somador Completo

O meio somador possibilita efetuar a soma de números binários com 1 algarismo. Mas o mundo real se faz necessário que esta soma seja efetuadas com um numero maior algarismo. Para satisfazer estas condições o circuito necessita de uma entrada de transporte proveniente de uma saída de transporte anterior. Para melhor compreensão, vamos analisar o caso da soma a seguir:

Desta forma a tabela verdade ficaria do seguinte modo:

A	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Colocando no mapa de Karnaugh, teremos o esquema do circuito conhecido como Full Adder.

Ex1: Montar um sistema que some em BCD.

3.5.3 Meio Subtrator

Vamos fazer um flashback no assunto para podermos montar as tabelas verdades equivalentes.

0-0=0
0-1=1 e empresta 1
1-0=1
1-1=0

Vamos montar a tabela verdade de uma subtração de dois números binários de 1 algarismo.

A	B	Saída (S)	Transporte (Ts)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Assim de forma análoga ao o circuito meio somador teremos a seguinte simplificação:

S=A exclusivo ou B
Ts= $\bar{A} + B$

3.5.4 Subtrator Completo

Novamente, o meio somador nos permite efetuar a subtração de apenas números com 1 algarismo. Para satisfazer uma subtração completa, devera ser inserida novamente uma entrada de transporte para que se possa montar tal circuito.
Assim teremos a seguinte tabela verdade:

A	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Novamente aplicando Karnaugh teremos o circuito simplificado do Subtrator Completo.

Ex: Montar um sistema que efetue a subtração de 2 números binários codificados em BCD.

Dispositivos de Memória

3.6 Memórias

A principal vantagem dos sistemas digitais sobre os analógicos é a capacidade de armazenar, facilmente, grandes quantidades de informação e/ou dados por períodos longos ou curtos de tempo. Esta capacidade de memória é o que torna os sistemas digitais tão versáteis e adaptáveis às diversas situações. Por exemplo, em um computador digital, a memória principal armazena instruções que informam ao computador o que fazer sob qualquer circunstância possível, de modo que o computador realizará sua tarefa com um mínimo de intervenção humana.

Vamos estudar os tipos mais comuns desses dispositivos e sistema de memória. Já estamos bem familiarizados com o *flip-flop*, que é um dispositivo eletrônico de memória. Também analisamos como grupos de FFs, chamados de *registradores*, podendo ser utilizados para armazenar informação e como esta informação pode ser transferida para outros lugares. Registradores são elementos de memória de alto desempenho que são muito usados nas ações internas de um computador digital, no qual a informação digital está sendo continuamente transferida de um local para outro. Os avanços na tecnologia LSI (*Large Scale Integration*) e VLSI (*Very Large Scale Integration*) foram possível a obtenção de um grande número de FFs,único chip, organizados em vários arranjos de memória.

Então as memórias são os dispositivos que armazenam informações, essas por sua vez codificadas, digitalmente, através de um código binário qualquer. Essas informações podem ser números, letras, caracteres quais quer, comandos de operações, endereços ou ainda qualquer outro tipo de dado.

Essas informações, armazenam dados para endereçamento, programação e para constituir o conjunto de funções internas para a funcionalidade do próprio sistema. Outra tipo de aplicação consiste em utilizá-las para executarem quaisquer funções de circuitos combinacionais, e ainda, com o auxílio de contadores comuns e conversores, gerar formas de onda de diversas maneiras de modo mais simples.

3.6.1 Classificação das Memórias

Antes de estudarmos os diversos tipos de memórias, vamos conhecer sua classificação. Podemos classificar as memórias em vários ítens diferentes. A seguir, vamos relacionar os principais:

- Acesso
- Volatilidade
- Escrita/Leitura ou apenas de leitura
- Tipo de armazenamento

Vamos analisar cada ítem:

1. **Acesso:**
As memórias armazenam informações em lugares denominados **localidade de memória**. Cada um das localidades de memória possui um conjunto de bits que nos permite o seu acesso, a esse conjunto de bits damos o nome de **endereço**. Esse conceito é de fácil compreensão, pois como o próprio nome diz, o conjunto de bits representa o endereço da localidade onde está armazenada uma informação.

O **tempo de acesso** de uma memória é o tempo necessário desde a entrada de um endereço até o momento em que a informação apareça na saída. Para as memórias de escrita/leitura é também o tempo necessário para a informação ser gravada.

Podemos ter acesso a uma dada localidade de memória de duas maneiras diferentes:

- acesso seqüencial;
- acesso aleatório.

2. **Volatilidade:**

Quanto à volatilidade, as memórias podem ser voláteis ou não voláteis. As memórias voláteis são aquelas que ao ser'cortada a alimentação perdem as informações armazenadas. São memórias feitas, geralmente, a partir de semicondutores e na maioria das vezes, possuem como elemento de memória o flip-flop. Um exemplo típico, já citado, é o da memória RAM. As memórias não voláteis são aquelas que mesmo sem alimentação continuam com as informações armazenadas. Dentre essas se destacam as memórias magnéticas e as eletrônicas: **ROM**, **PROM** e **EPROM**.

3. **Memórias de escrita/leitura ou memórias apenas de leitura:**

As memórias de escrita/leitura são aquelas que permitem acesso a uma localidade qualquer para escrevermos a informação desejada, além disso, permitem o acesso também para a leitura do dado.

As memórias RAM também se enquadraram nessa situação. As memórias apenas de leitura, como o próprio nome diz, são.aquelas em que a informação. é fixa, só podendo efetuar-se a leitura. São também conhecidas como ROM (*Read Only Memory*). A análise desses tipos de memórias será feita mais adiante através dos seminários.

4. **Tipos de armazenamento:**

Quanto ao tipo de armazenamento as memórias classificam-se em estáticas e dinâmicas.

As memórias de armazenamento estático são aquelas em que uma vez inserido o dado numa dada localidade, este lá permanece.

As memórias de armazenamento dinâmico são aquelas em que necessitamos inserir a informação de tempos em tempos, pois de acordo com as características de seus elementos internos perdem essas informações após um determinado tempo.

As memórias de armazenamento estático apresentam a vantagem de possuir uma utilização da maneira mais fácil que as dinâmicas.

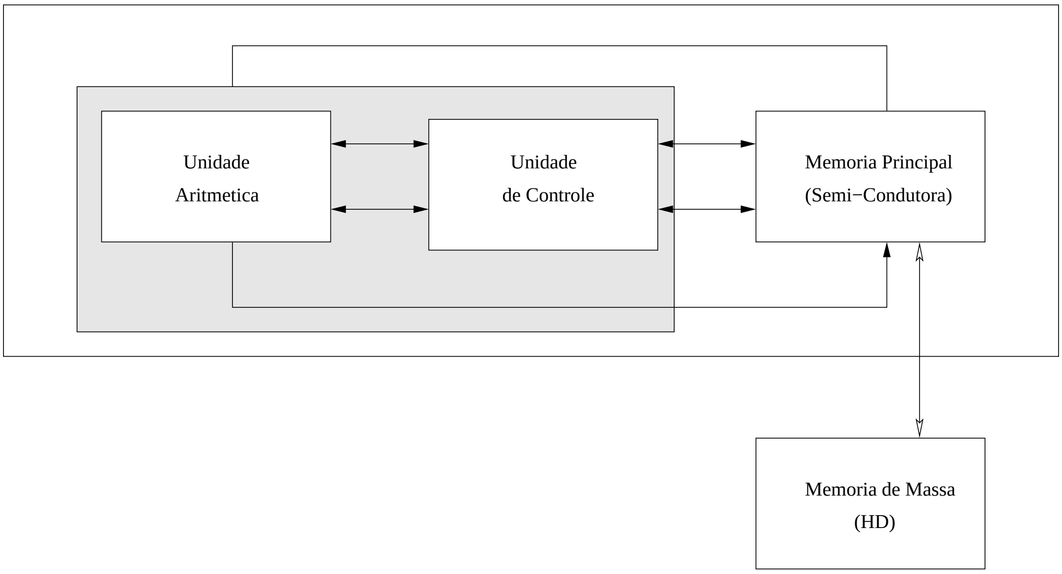


Figura 3.10: Arquitetura de um computador

3.7 Terminologia

O estudo dos sistemas e dos dispositivos de memória está repleto de termos. É de grande valia que você possa compreender o significado de alguns termos mais básicos, que são eles:

1. **Célula de memória**
Um dispositivo ou circuito elétrico utilizado para armazenar um único bit (0 ou 1). Exemplos de célula de memoria incluem: um flip-flop, um capacitor carregado e um pequeno local numa fita ou disco magnético.

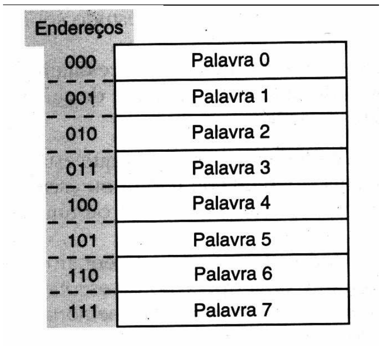
2. **Palavra de memória:**
Um grupo de bits (células) em uma memória que representa instruções ou dados de algum tipo. Por exemplo, um registrador de oito FFs pode ser considerado uma memória que esta armazenando uma palavra de 8 bits. Os tamanhos de palavra nos computadores modernos variam tipicamente de 4 a 64 bits, dependendo do porte do computador.

3. **Byte:**
Um termo especial usado para um grupo de oito bits. Um byte sempre e constituído de 8 bits. Tamanhos de palavra podem ser expressos em bytes assim como em bits. Por exemplo, uma palavra de 8 bits e também uma palavra de um byte; uma palavra de 16 bits tem dois bytes, e assim por diante.

4. **Capacidade:**
Uma maneira de especificar quantos bits podem ser armazenados em um determinado dispositivo de memória ou num sistema de memória completo. Para ilustrar, suponha que temos uma memória capaz de armazenar 4.096 palavras de 20 bits. Isto representa uma capacidade total de 81.920 bits. Poderíamos também expressar essa capacidade de memória como 4.096 X 20. Quando representada desse modo, o primeiro número (4.096) é o número de palavras, e o segundo número (20) é o número de bits por palavra (tamanho da palavra). O número de palavras em uma memória freqüentemente é um múltiplo de 1.024. É comum usar a designação "1K"para representar $1.024 = 2^{10}$ quando nos referimos a capacidade de memória. Logo, uma memória com uma capacidade de armazenamento de 4K X 20 e na verdade uma memória de 4.096 X 20. O desenvolvimento de memórias maiores trouxe a designação "1M"ou "1 mega"para representar $2^{20} = 1.048.576$. Assim, uma memória que possui uma capacidade de 2M X 8 tem na verdade uma capacidade de 2.097.152 x 8. A designação "giga"se refere a $2^{30} = 1.073.741.824$.

5. **Densidade:**
Um outro termo para capacidade. Quando dizemos que um dispositivo de memória tem uma densidade maior do que um outro, queremos dizer que ele pode armazenar mais bits no mesmo espaço, ou seja ele é mais de denso.

6. **Endereço:**
É um número que identifica a posição de palavra na memória. Cada palavra armazenada em um dispositivo ou sistema de memória possui um endereço único. Endereços sempre existem num sistema digital como um número binário, embora, por conveniência, números em octal, hexadecimal e decimal sejam freqüentemente utilizados para representar esses endereços.



Endereços	
000	Palavra 0
001	Palavra 1
010	Palavra 2
011	Palavra 3
100	Palavra 4
101	Palavra 5
110	Palavra 6
111	Palavra 7

Figura 3.11: Tabela de endereços de memória

A figura 3.11 ilustra uma pequena memória constituída de oito palavras. Cada uma destas oito palavras tem um endereço específico representado por um número de três bits que varia de 000 até 111. Sempre que nos referimos a uma posição específica na memória, utilizamos seu código de endereço para identificá-la.

7. **Operação de Leitura:**
Operação na qual a palavra binária armazenada numa determinada posição (endereço) de memória é detectada e então transferida para outro dispositivo. Por exemplo, se desejamos utilizar a palavra 4 da memória da figura anterior para algum propósito, devemos realizar uma operação de leitura no endereço 100. A operação de leitura freqüentemente é chamada de operação de busca, pois a palavra está sendo buscada da memória. Utilizaremos os dois termos indistintamente.

8. **Operação de Escrita:**
Operação na qual uma nova palavra é colocada numa determinada posição de memória. Também é chamada de operação de armazenamento. Sempre que uma nova palavra é escrita numa posição de memória, ela substitui a palavra que estava previamente armazenada lá.

9. **Tempo de Acesso:**
Uma medida da velocidade de operação de um dispositivo de memória. É o tempo necessário para realizar uma operação de leitura. Mais especificamente, é o tempo entre a memória receber uma nova entrada de endereço e os dados se tornarem disponíveis na saída da memória. O símbolo t_{Acc} é utilizado para tempo de acesso.

10. **Memória Volátil:**
Qualquer tipo de memória que necessita da aplicação de energia para poder armazenar informação. Se a energia elétrica é removida, todas as informações armazenadas na memória

são perdidas. Muitas das memórias semicondutoras são voláteis, enquanto todas as memórias magnéticas são não-voláteis, o que significa que elas podem armazenar informação sem energia elétrica.

11. **Memória de Acesso Aleatório (RAM -Random Access Memory):**
Memória na qual a posição física real de uma palavra da memória não tem efeito sobre o tempo necessário para ler ou escrever nesta posição. Em outras palavras, o tempo de acesso é o mesmo para qualquer endereço na memória. A maioria das memórias semicondutoras é de acesso aleatório.

12. **Memória de Acesso Seqüencial (SAM -Sequence Access Memory)**
Um tipo de memória no qual o tempo de acesso não é constante mas varia dependendo do endereço. Uma determinada palavra armazenada é encontrada percorrendo todos os endereços até que o endereço desejado seja alcançado. Isto produz tempos de acesso que são muito maiores do que os das memórias de acesso aleatório. Um exemplo de dispositivo de memória de acesso seqüencial é uma fita magnética. Para ilustrar a diferença entre SAM e RAM, considere a situação na qual você gravou 60 minutos de música numa fita cassete de audio. Quando desejar alcançar uma música em particular, você terá que retroceder ou avançar a fita até a encontrar. O processo é relativamente lento, e o tempo necessário depende de onde a música desejada está gravada na fita. Isto é SAM, já que você percorreu através das informações registradas até encontrar o que estava procurando. A contrapartida RAM para isso seria um CD ou MD de audio, no qual você pode rapidamente selecionar qualquer música informando o código apropriado, e ele gasta aproximadamente o mesmo tempo, não importando a música selecionada. As memórias de acesso seqüencial são utilizadas onde os dados a serem acessados sempre vêm numa longa seqüência de palavras sucessivas. A memória de vídeo, por exemplo, deve fornecer seu conteúdo na mesma ordem repetidamente para manter a imagem na tela.

13. **Memória de Leitura e Escrita (RWM -Read/Write Memory):**
Qualquer memória que possa.ser lida ou escrita de maneira igualmente fácil.

14. **Memória Somente de Leitura (ROM - Read-Only Memory):**
Uma vasta classe de memórias semicondutoras, projetadas para aplicações nas quais a razão entre as operações de leitura e escrita é muito alta. Tecnicamente, uma ROM pode ser escrita (programada) apenas uma vez, e esta operação normalmente é realizada na fábrica. Depois disso, as informações podem ser somente lidas da memória. Outros tipos de ROM são na verdade RMM (read-mostly memories), nas quais se pode escrever mais de uma vez; porém a operação de escrita é mais complicada do que a de leitura, e não é realizada freqüentemente. Os vários tipos de ROM serão apresentadas em forma de seminários. Todas as ROMs são não-voláteis e armazenam dados quando a energia é removida.

15. **Dispositivos de Memória Estática:**
Dispositivos de memória semicondutora nos quais os dados permanecem armazenados enquanto a energia está presente, sem a necessidade de reescrever periodicamente os dados. na memória.

16. **Dispositivos de Memória Dinâmica:**
Dispositivos de memória semicondutora nos quais os dados não permanecem armazenados, mesmo com a energia presente, a menos que os dados sejam periodicamente reescritos na memória. Esta última operação é denominada refresh.

17. **Memória Principal:**
Também chamada de memória de trabalho do computador. Ela armazena instruções e dados que a CPU está acessando no momento. É a memória mais rápida num computador e sempre é uma memória semicondutora.

18. **Memória Auxiliar:**
Também chamada de memória de massa porque ela armazena grandes quantidades de informação externamente à memória principal. É mais lenta do que a memória principal e sempre é não-volátil. Discos magnéticos e CDs são dispositivos comuns de memória auxiliar.

3.8 Princípios de Operação da Memória

Embora cada tipo de memória seja diferente na sua operação interna, certos princípios básicos são comuns a todas elas.

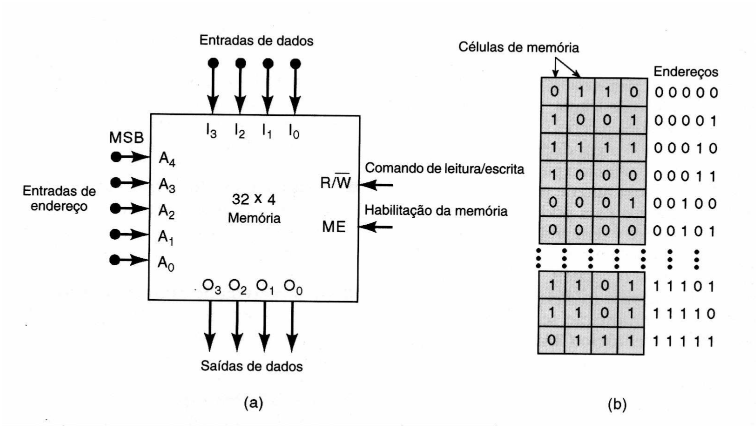


Figura 3.12: Bloco de memória

Todos os dispositivos de memória necessitam de diversos tipos diferentes de linhas de entrada e de saída para realizar as seguintes funções:

1. Selecionar o endereço na memória que está sendo acessado para uma operação de leitura ou escrita;

2. Selecionar uma operação de leitura ou escrita que será realizada;

3. Fornecer os dados de entrada a serem armazenados na memória durante uma operação de escrita;

4. Manter os dados de saída vindos da memória durante uma operação de leitura;

5. Habilitar (ou desabilitar) a memória de modo que ela responda(ou não) às entradas de endereçamento e ao comando de leitura/escrita.

3.8.1 Entradas de Endereço

Utilizando o bloco anterior como exemplo, a memória armazena 32 palavras, ela tem 32 posições de armazenamento diferentes, e portanto possui 32 endereços binários diferentes, variando de 00000 até 11111 (0 a 31 em decimal). Logo, existem cinco entradas de endereço, A_0 até A_4. Para acessar cada uma das posições de memória para uma operação de leitura ou escrita, o código de endereçamento de cinco bits para essa posição é aplicado nas entradas de endereço. De um modo geral, N entradas de endereço são necessárias para uma memória que possui uma capacidade de 2^N palavras.
Podemos visualizar a memória da figura como um arranjo de 32 registradores, no qual cada registrador guarda uma palavra de quatro bits, conforme mostra o mesmo. Cada posição é mostrada contendo

quatro células de memória que guardam 1s ou 0s, que formam a palavra de dados armazenada nesta posição. Vejamos o seguinte exemplo, a palavra 0110 está armazenada no endereço 00000, a palavra de dados 1001 está armazenada no endereço 00001, e assim por diante.

3.8.2 A Entrada R/$\overline{\text{W}}$

Esta entrada controla qual operação deve ser realizada na memória: leitura (R - read) ou Escrita (W - write). A entrada é identificada por R/$\overline{W}$, e, como não existe a barra sobre R, isto indica que a operação de leitura ocorre quando R/$\overline{W}$=1. A barra sobre W indica que a operação de escrita acontece quando R/$\overline{W}$=0. Outros identificadores (nomenclaturas de outros autores)são usados freqüentemente para essa entrada. Dois dos mais comuns são $\overline{W}$(escrita) e $\overline{WE}$ (write enable-habilitação de escrita). Novamente, a barra indica que a operação de escrita ocorre quando a entrada está em BAIXO. Fica subentendido que a operação de leitura ocorre para nível alto.

3.8.3 Habilitação da Memória

Muitos sistemas de memória tem algum modo de desabilitar completamente uma parte ou toda a memória, de modo que ela não possa responder às outras entradas. Isto é representado na figura anterior pela entrada ME, embora ela possa ter nomes diferentes nos vários tipos de memória, tais como *chip enable (CE)* ou *chip select (CS)*. Na figura, ela é mostrada como uma entrada ativa em ALTO que habilita a memória, de modo que ela não responderá às entradas de endereço e de R/$\overline{W}$. Esse tipo de entrada é útil quando vários módulos de memória são combinados para formar uma memória maior.

3.8.4 Exercícios

1ª **Exercício:** Um certo chip de memória semicondutora é especificado como 4K X 8. Quantas palavras podem ser armazenadas neste chip? Qual é o tamanho da palavra? Quantos bits neste chip pode armazenar no total?

2ª **Exercício:** Qual das memórias armazena mais bits: uma memória de 5M X 8 ou uma memória que armazena 2M palavras com um tamanho de palavra de 16 bits?

3ª **Exercício:** Descreve as condições de cada entrada e saída quando o conteúdo da posição cujo endereço é 00100 deve ser lido.

4ª **Exercício:** Descreva as condições de cada entrada e saída quando a palavra 1110 deve ser escrita na posição de endereço 01101

5ª **Exercício:** Uma determinada memória tem uma capacidade de 4K X 8.

(a) Quantas linhas de entrada de dados e saída de dados ela tem?

(b) Quantas linhas de endereço ela tem?

(c) Qual é a sua capacidade em bytes?